CLAIMS

1. A memory controller, comprising:

at least one bus interface, each bus interface being for connection to at least one respective device for receiving memory access requests;

a memory interface, for connection to a memory device over a memory bus;

a plurality of buffers; and

control logic, for placing received memory access requests into a queue of memory access requests,

wherein, in response to a received memory access request requiring multiple data bursts over the memory bus, data from each of said multiple data bursts is stored in a respective buffer of said plurality of buffers.

2. A memory controller as claimed in claim 1, wherein, when returning data to the respective device from which a memory access request requiring multiple data bursts over the memory bus was received, data is read out from a first part of one of said buffers, then data is read out from at least one other of said buffers, then data is read out from a second part of said one of said buffers.

3. A memory controller as claimed in claim 1, wherein said plurality of buffers is located in said memory interface.

4. A memory controller as claimed in claim 1, wherein the or each bus interface comprises a respective plurality of buffers.

5. A memory controller as claimed in claim 1, wherein the control logic determines whether a received read access request is a wrapping request which requires multiple memory bursts, and, if so, the control logic allocates each of said memory bursts to a respective one of said buffers.

6. A memory controller as claimed in claim 1, wherein the memory controller is a SDRAM controller, and said memory interface is suitable for connection to a SDRAM memory device over said memory bus.

7. In a memory controller, comprising: at least one bus interface, each bus interface being for connection to at least one respective device for receiving memory access requests; a memory interface, for connection to a memory device over a memory bus; a plurality of buffers; and control logic, for placing received memory access requests into a queue of memory access requests, a method of retrieving data comprising:

in response to a received memory access request requiring multiple data bursts over the memory bus, storing data from each of said multiple data bursts in a respective buffer of said plurality of buffers.

8. A method as claimed in claim 7, further comprising, when returning data to the respective device from which a memory access request requiring multiple data bursts over the memory bus was received, reading data out from a first part of one of said buffers, then reading data out from at least one other of said buffers, then reading data out from a second part of said one of said buffers.

9. A method as claimed in claim 7, comprising storing said data from each of said multiple data bursts in a respective buffer in said memory interface.

10. A method as claimed in claim 7, comprising storing said data from each of said multiple data bursts in a respective buffer in said bus interface to which the respective device from which the memory access request was received.

11. A method as claimed in claim 7, comprising determining whether a received read access request is a wrapping request which requires multiple memory bursts, and, if so, allocating each of said memory bursts to a respective one of said buffers.

12. A method as claimed in claim 7, wherein the memory controller is a SDRAM controller, and said memory interface receives data from a SDRAM memory device over said memory bus in SDRAM bursts.

13. A programmable logic device, wherein the programmable logic device includes a memory controller, comprising:

at least one bus interface, each bus interface being for connection to at least one respective device formed within the programmable logic device for receiving memory access requests;

a memory interface, for connection to an external memory device over a memory bus;

a plurality of buffers; and

control logic, for placing received memory access requests into a queue of memory access requests, wherein, in response to a received memory access request requiring multiple data bursts over the memory bus, data from each of said multiple data bursts is stored in a respective buffer of said plurality of buffers.